SNA Basics VIII

Centrality and Prestige

@ Sean F. Everton

Place a header at the top of your script that tells you what you called it, what it accomplishes, etc.

```
##################################################
# What: Centrality and Prestige in R
# File: snab8.R
# Created: 02.28.29
# Revised: 07.03.18
##################################################
```

Data

When we use centrality to measure prestige, we typically do so with directed (rather than undirected) network data. Thus, for this exercise we will use directed network data collected by David Krackhardt from the managers of a high-tech company on the West Coast. At the time, the firm had existed for 10 years, produced high-tech machinery for other companies, and employed approximately 100 people of whom 21 were managers. The 21 managers are the actors in the dataset. Krackhardt gave each manager a roster of the names of the other managers and asked to check the other managers to whom they would go for advice at work and with whom they were friends. He also mapped the firm's offical hierarchy ("Reports To"). The Krackhardt dataset is one of the standard datasets that comes with most SNA software packages like UCINET and is commonly used to illustrate prestige. As far as I know, no religious data were collected on the managers, however. :-)

Setup

Clear the workspace each time before beginning.

```
rm(list=ls())
```

Set your working directory to where the data are, so you don't have to include the entire path when importing and exporting data, files, etc.

```
setwd("~/Dropbox/Networks and Religion (Book)/Website/Labs/SNA Basics 8")
```

Centrality and Prestige in *statnet*

We need to first load the libraries we plan to use. The *sna* and *network* libraries are part of the *statnet* package. Because *igraph* and *sna* conflict with one another, we can't load them at the same time. Well, we can, but it only causes unnecessary trouble. There is a workaround, but it's just as easy to attach and detach the two libraries.

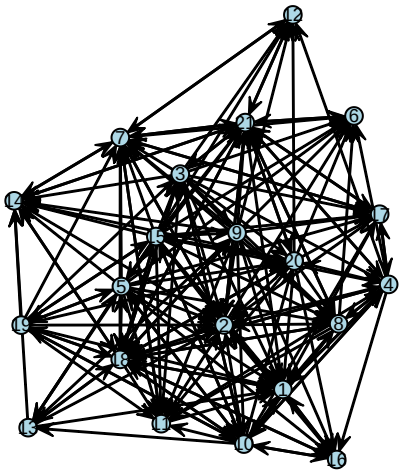
```
library(sna)
library(network)
```

Import the Advice and Friendship networks using the Pajek files

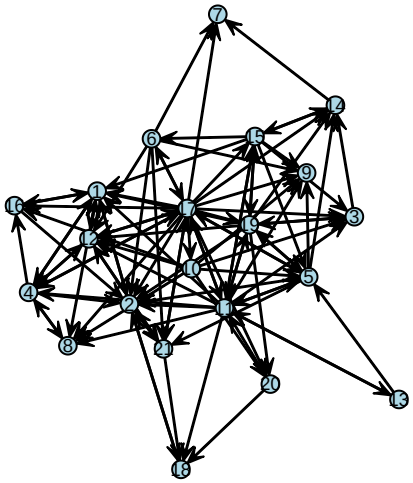
```
advice.net <- as.network(read.paj("ADVICE.net"),directed=TRUE)
friendship.net <- as.network(read.paj("FRIENDSHIP.net"),directed=TRUE)
reports.net <- as.network(read.paj("REPORTS_TO.net"),directed=TRUE)
```

Some initial plots to see how the networks look; note the difference between the official (reports to) and unofficial (advice and friendship) liness of influence

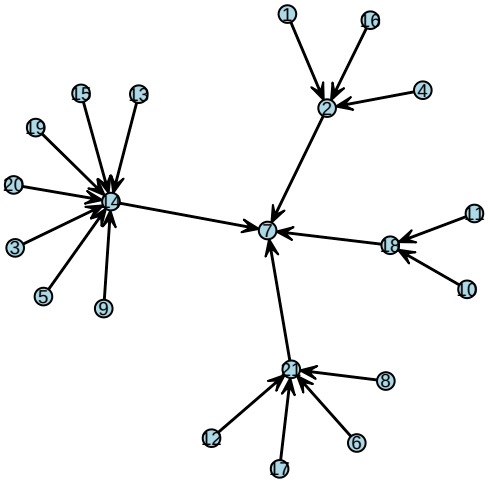
```
coorda <- gplot(advice.net,label=network.vertex.names(advice.net),label.col="black",
                label.pos=5,label.cex=0.6,vertex.col="light blue",usearrows=TRUE,
                gmode="digraph")
```



```
coordf <- gplot(friendship.net,label=network.vertex.names(friendship.net),label.col="black",
                label.pos=5,label.cex=0.6,vertex.col="light blue",usearrows=TRUE,
                gmode="digraph")
```



```
coordr <- gplot(reports.net,label=network.vertex.names(reports.net),label.col="black",
                label.pos=5,label.cex=0.6,vertex.col="light blue",usearrows=TRUE,gmode="digraph")
```



Calculate in-degree and out-degree for all three networks

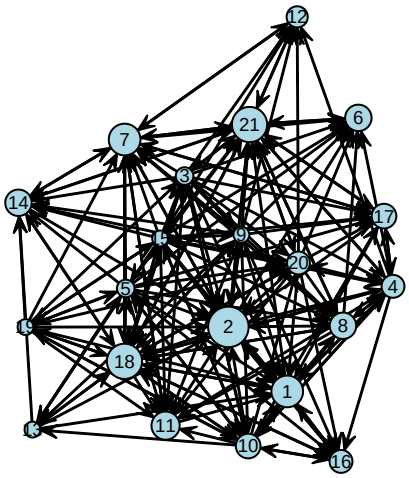
```
advice.indeg <- degree(advice.net,gmode="digraph",cmode="indegree")
advice.outdeg <- degree(advice.net,gmode="digraph",cmode="outdegree")

friendship.indeg <- degree(friendship.net,gmode="digraph",cmode="indegree")
friendship.outdeg <- degree(friendship.net,gmode="digraph",cmode="outdegree")

reports.indeg <- degree(reports.net,gmode="digraph",cmode="indegree")
reports.outdeg <- degree(reports.net,gmode="digraph",cmode="outdegree")
```

statnet also has a "prestige" function, which allows you to access a variety of prestige measures (including in and out-degree)

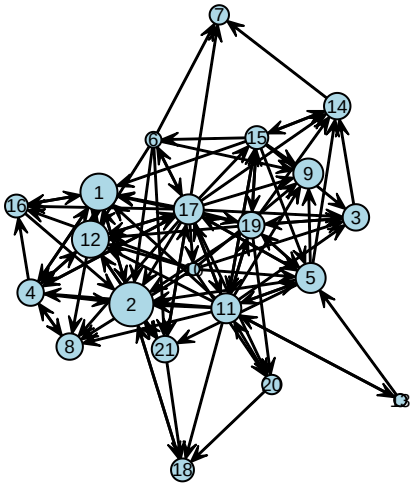
```
advice.in <- prestige(advice.net,cmode="indegree")
friendship.in <- prestige(friendship.net,cmode="indegree")
reports.in <- prestige(reports.net,cmode="indegree")

# Check correlation (should be 1.0)
cor(advice.indeg,advice.in)
## [1] 1
cor(friendship.indeg,friendship.in)
## [1] 1
cor(reports.indeg,reports.in)
## [1] 1
```

Now, let's vary node size of plots by indegree.

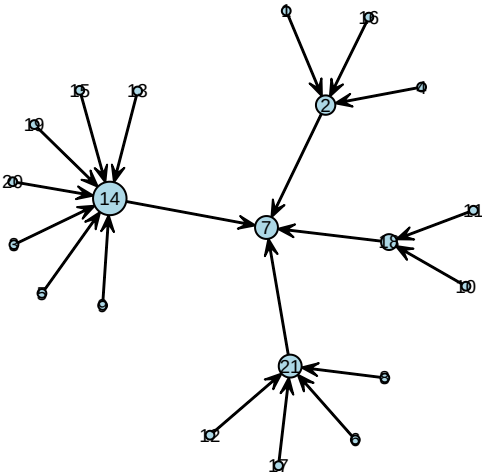
```
gplot(advice.net,label=network.vertex.names(advice.net),label.col="black",label.pos=5,
      label.cex=0.6,vertex.col="light blue",usearrows=TRUE,gmode="digraph",
      vertex.cex=((advice.indeg/10)+.5),coord=coorda)
```



```
gplot(friendship.net,label=network.vertex.names(friendship.net),label.col="black",label.pos=5,
      label.cex=0.6,vertex.col="light blue",usearrows=TRUE,gmode="digraph",
      vertex.cex=((friendship.indeg/5)+.5),coord=coordf)
```



```
gplot(reports.net,label=network.vertex.names(reports.net),label.col="black",label.pos=5,
      label.cex=0.6,vertex.col="light blue",usearrows=TRUE,gmode="digraph",
      vertex.cex=((reports.indeg/5)+.5),coord=coordr)
```



The prestige function can estimate more than in- and out-degree, such as eigenvector, input domain, and proximity prestige.

```
# Eigenvector
advice.eig <- prestige(advice.net,cmode="eigenvector")
friendship.eig <- prestige(friendship.net,cmode="eigenvector")

# Input Domain
advice.dom <- prestige(advice.net,cmode="domain")
friendship.dom <- prestige(friendship.net,cmode="domain")

# Proximity prestige
advice.prox <- prestige(advice.net,cmode="domain.proximity")
friendship.prox <- prestige(friendship.net,cmode="domain.proximity")
```

Create a data frame to dispaly the prestige scores for the advice and friendship networks (we'll add authorities once we move to *igraph*)

```
manager <- network.vertex.names(advice.net)
aprestige <- data.frame(manager,advice.in,advice.eig,advice.dom,advice.prox)
fprestige <- data.frame(manager,friendship.in,friendship.eig,friendship.dom,friendship.prox)

colnames(aprestige) <- c("Manager","Indegree","Eigenvector","Input Domain","Proximity Prestige")
colnames(fprestige) <- c("Manager","Indegree","Eigenvector","Input Domain","Proximity Prestige")

aprestige
##    Manager Indegree Eigenvector Input Domain Proximity Prestige
## 1        1       13  0.23975517           20          0.6666667
## 2        2       18  0.40340470           20          0.9090909
## 3        3        5  0.12353507           20          0.5555556
## 4        4        8  0.20856936           20          0.6250000
## 5        5        5  0.08821983           20          0.5000000
## 6        6       10  0.25022117           20          0.6666667
## 7        7       13  0.30957894           20          0.7407407
## 8        8       10  0.22390781           20          0.6666667
## 9        9        4  0.07350524           20          0.4878049
## 10      10        9  0.16536656           20          0.5714286
## 11      11       11  0.22418114           20          0.6666667
## 12      12        7  0.16352716           20          0.5882353
## 13      13        4  0.07961599           20          0.4878049
## 14      14       10  0.20635411           20          0.6666667
## 15      15        4  0.08893586           20          0.4878049
## 16      16        8  0.16397876           20          0.5714286
## 17      17        9  0.19425173           20          0.6451613
## 18      18       15  0.31459137           20          0.8000000
## 19      19        4  0.07961599           20          0.4878049
## 20      20        8  0.17446162           20          0.6060606
## 21      21       15  0.37109164           20          0.8000000
fprestige
##    Manager Indegree Eigenvector Input Domain Proximity Prestige
## 1        1        8  0.39231915           18                  0
## 2        2       10  0.42552937           18                  0
## 3        3        5  0.16144491           18                  0
## 4        4        5  0.27194316           18                  0
## 5        5        6  0.16834101           18                  0
```

```
## 6        6        2  0.08089575          18                0
## 7        7        3  0.10194939          19                0
## 8        8        5  0.22611993          18                0
## 9        9        6  0.17495152          19                0
## 10      10        1  0.04988790          18                0
## 11      11        6  0.17589765          18                0
## 12      12        8  0.32713825          18                0
## 13      13        1  0.03482824          18                0
## 14      14        5  0.18203729          18                0
## 15      15        4  0.15660303          18                0
## 16      16        4  0.19129169          18                0
## 17      17        6  0.25195542          18                0
## 18      18        4  0.18717284          18                0
## 19      19        5  0.18102254          18                0
## 20      20        3  0.09560883          18                0
## 21      21        5  0.24826784          18                0
```

Centrality and Prestige in *igraph*

Let's move to *igraph* and see how to do things there. We need to load *igraph* and detach *sna*. We'll use *intergraph* to tranform the networks into *igraph* objects

```
library(igraph)
library(intergraph)
detach("package:sna", unload=TRUE)

advice.ig <- asIgraph(advice.net)
friendship.ig <- asIgraph(friendship.net)
reports.ig <- asIgraph(reports.net)
```

Calculate in-degree and then correlate with indegree caluclated using *statnet* - the correlations should = 1.0 (and they do)

```
advice.igindeg <- degree(advice.ig,mode = c("in"))
friendship.igindeg <- degree(friendship.ig,mode = c("in"))
reports.igindeg <- degree(reports.ig,mode = c("in"))

cor(advice.igindeg,advice.indeg)
## [1] 1
cor(friendship.igindeg,friendship.indeg)
## [1] 1
cor(reports.igindeg,reports.indeg)
## [1] 1
```

Hubs (out-degree eigenvector) and Authorities (in-degree eigenvector)

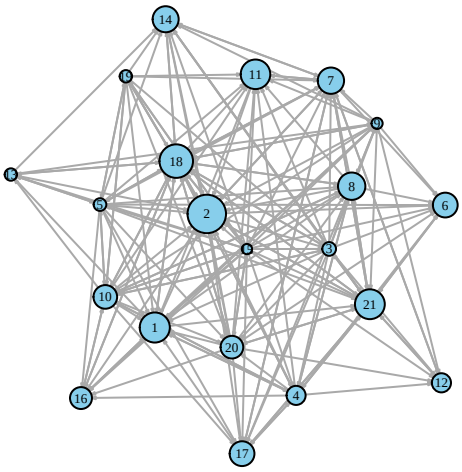
```
advice.hubs <- hub.score(advice.ig,scale=FALSE)
friendship.hubs <- hub.score(friendship.ig,scale=FALSE)

advice.auth <- authority.score(advice.ig,scale=FALSE)
friendship.auth <- authority.score(friendship.ig,scale=FALSE)
```

Create a table of prestige scores for the advice and friendship networks

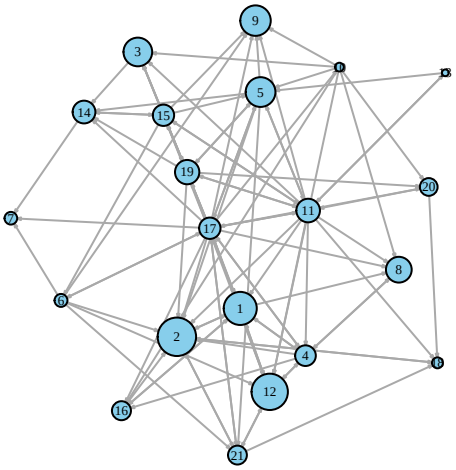
```
aprestige$Authority <- advice.auth$vector
fprestige$Authority <- friendship.auth$vector

aprestige
##    Manager Indegree Eigenvector Input Domain Proximity Prestige  Authority
## 1        1       13  0.23975517           20          0.6666667 0.27715635
## 2        2       18  0.40340470           20          0.9090909 0.35425017
## 3        3        5  0.12353507           20          0.5555556 0.12621843
## 4        4        8  0.20856936           20          0.6250000 0.17573188
## 5        5        5  0.08821983           20          0.5000000 0.11690530
## 6        6       10  0.25022117           20          0.6666667 0.22815619
## 7        7       13  0.30957894           20          0.7407407 0.24243281
## 8        8       10  0.22390781           20          0.6666667 0.25179672
## 9        9        4  0.07350524           20          0.4878049 0.10265888
## 10      10        9  0.16536656           20          0.5714286 0.21775410
## 11      11       11  0.22418114           20          0.6666667 0.27226779
## 12      12        7  0.16352716           20          0.5882353 0.17628866
## 13      13        4  0.07961599           20          0.4878049 0.11426369
## 14      14       10  0.20635411           20          0.6666667 0.23978225
## 15      15        4  0.08893586           20          0.4878049 0.09467238
## 16      16        8  0.16397876           20          0.5714286 0.20201014
## 17      17        9  0.19425173           20          0.6451613 0.22836693
## 18      18       15  0.31459137           20          0.8000000 0.30866989
## 19      19        4  0.07961599           20          0.4878049 0.11426369
## 20      20        8  0.17446162           20          0.6060606 0.20861102
## 21      21       15  0.37109164           20          0.8000000 0.27475792
fprestige
##    Manager Indegree Eigenvector Input Domain Proximity Prestige  Authority
## 1        1        8  0.39231915           18                   0 0.30459884
## 2        2       10  0.42552937           18                   0 0.35393230
## 3        3        5  0.16144491           18                   0 0.26406201
## 4        4        5  0.27194316           18                   0 0.19036195
## 5        5        6  0.16834101           18                   0 0.27414262
## 6        6        2  0.08089575           18                   0 0.11913258
## 7        7        3  0.10194939           19                   0 0.11658203
## 8        8        5  0.22611993           18                   0 0.23703212
## 9        9        6  0.17495152           19                   0 0.28033987
## 10      10        1  0.04988790           18                   0 0.08029898
## 11      11        6  0.17589765           18                   0 0.21723668
## 12      12        8  0.32713825           18                   0 0.33466738
## 13      13        1  0.03482824           18                   0 0.06191933
## 14      14        5  0.18203729           18                   0 0.20952877
## 15      15        4  0.15660303           18                   0 0.19618125
## 16      16        4  0.19129169           18                   0 0.17511279
## 17      17        6  0.25195542           18                   0 0.19658051
## 18      18        4  0.18717284           18                   0 0.10053613
## 19      19        5  0.18102254           18                   0 0.22390147
## 20      20        3  0.09560883           18                   0 0.16330908
## 21      21        5  0.24826784           18                   0 0.17375069
```

Two last plots; the advice and friendship networks with node size varying by authority scores

```
plot(advice.ig,vertex.size=advice.auth$vector*50,vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue",edge.arrow.size=.1)
```




```
plot(friendship.ig,vertex.size=friendship.auth$vector*50,vertex.label.cex=.4,
     vertex.label.color="black",vertex.color="Sky Blue",edge.arrow.size=.1)
```



That's all for now